

In the Claims

1. **(Currently Amended)** A complex order processing system comprising:
 - a plurality of business service methods;
 - user interface instructions **operable configured** to[:]]
 - generate a customer portal view comprising
 - a list of products and services associated with an account,
 - a list of quotes associated with the account,
 - a list of orders associated with the account, and
 - a plurality of user-selectable options[[:]] **, and**
 - receive input from a user, **wherein**
 - the input comprises a selection of a least a selected option of the plurality of user-selectable options;** and
 - a plurality of workflow processes, wherein
 - at least one of the plurality of workflow processes is invoked ~~when a user selects one~~ **in response to the selection of the at least the option** of the plurality of user-selectable options, and
 - the at least one of the plurality of workflow processes ~~invokes~~ **is configured to invoke** at least one of the plurality of business service methods to perform ~~functions~~ **a function** associated with the selected option.
2. **(Original)** The complex order processing system of claim 1, further comprising a plurality of complex objects, wherein each complex object is based on a property set of at least one of: a complex asset, a quote, and an order; and the business service methods include at least one of:
 - an Apply Method to create a hybrid asset/order that simulates the future configuration of the complex asset by overlaying unprocessed items and attributes from open orders on one of the plurality of complex objects that is based on the complex asset;
 - a Delta Method to generate a delta complex object that represents the difference between at least two of the plurality of complex objects;

- a Trim Method to create a new complex object by eliminating all items that meet predefined trim criteria from one of the plurality of complex objects;
- a Logical Delete Method to generate a transformed delta order complex object by transforming items in one of the plurality of complex objects with an action code “Delete” to “Update” and changing status of the transformed items to “Inactive”;
- an Explode Method to copy items in one of the plurality of complex objects with quantity greater than one, and generate a delta quote complex object with a corresponding number of copies of the item with quantity equal to one;
- a Fully Exploded Method to determine whether the items in one of the plurality of complex objects have a quantity greater than one;
- a Field Delta Method to set the action code of a parent quote item to “Update” when a specified field is updated;
- a Set Effectivity Dates Method to generate an “Effective End Date”, an “Effective Start Date”, and a “Completed Date”;
- a Set Action Method to set the action code of an item in one of the plurality of complex objects;
- a Save Object Method to receive a key and a property set hierarchy that represents one of the plurality of complex objects, and saves the one of the plurality of complex objects in a string to object map using the key;
- a Get Object Method to receive the key and return the one of the plurality of complex objects that corresponds to the key;
- a Reset Method to delete all saved complex objects from the map;
- a Module Licensed Method to determine whether the complex order processing system is licensed for a particular computer system;
- an Update Order Status Method to check a Status field of a root item and all child items in the hierarchy of one of the plurality of complex objects based on a complex asset to determine whether the Status field is set to “Complete”, “Failed” or NULL;
- a Merge Method to copy all items from one of the plurality of complex objects to another of the plurality of complex objects;

a Set Field Method to enumerate through the item hierarchy of one of the plurality of complex objects, find a named field of each item, and set the named field to a specified value; and

a Create Empty Complex Object Method to create an empty complex object based on one of the plurality of complex objects.

3. (Original) The complex order processing system of claim 2, wherein the workflow processes include at least one of:

a New Products and Services Process to create a quote;

a Modify Products and Services Process to modify a complex asset in a quote;

a Disconnect Products and Services Process to disconnect a complex asset;

a Profile Process to retrieve account information that corresponds to a quote;

an Edit Quotes Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, and allow the user to make changes to the configuration of the complex asset and update a quote;

a Supplemental Orders Process to create a new order by allowing the user to revise a pending order;

an Edit Order Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, allow the user to make changes to the configuration of the complex asset, and update an order;

a Quote to Order Process to convert a quote to a sales order;

an Update Opportunity Process to create revenue items for all quote line items;

a Submit Order Process to associate account information with each line item in an order and set the status of the order to "open";

an Ungroup Quotes Process to create multiple copies of line items in a quote that have quantities greater than one;

an Apply Completed Sales Order Line Item To Service Profile Process to update an asset table for a complex asset that has been provisioned; and

a Move Process to transfer a customer's complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection.

4. **(Currently Amended)** The complex order processing system of claim 3, wherein the New Products and Services Process is **operable configured** to:

- retrieve a selected quote;
- create a new quote if no quote is selected;
- designate the billing and service accounts; and
- display the active quote.

5. **(Currently Amended)** The complex order processing system of claim 3, wherein the Modify Products and Services Process is **operable configured** to:

- retrieve the quote that is selected;
- create the quote if no quote is selected;
- allow the user to make changes to the currently requested state of the complex asset;
- apply changes specified by all open and pending orders related to the complex asset to
 - simulate the state of the service product;
- return an empty asset when the open and pending order disconnects a root component of
 - the complex asset;
- allow the user to change the configuration of the complex asset when the complex asset is
 - customizable;
- update the quote with the change between the previously requested state of the complex
 - asset and the newly requested state of the complex asset; and
- store the updated quote.

6. **(Currently Amended)** The complex order processing system of claim 3, wherein the Disconnect Products and Services Process is **operable configured** to:

- retrieve the quote that is selected;
- create the quote if no quote is selected; and
- allows the user to disconnect the complex asset, and create a quote line item with the
 - complex asset disconnected.

7. **(Currently Amended)** The complex order processing system of claim 3, wherein the Edit Quotes Line Items Process is **operable configured** to:

- determine whether the status of the action code of an order line item is “Add”;

generate the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;
apply the current quote line item to the complex asset;
display the currently requested state of the complex asset and allow the user to reconfigure the complex asset; and
generate the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset.

8. **(Currently Amended)** The complex order processing system of claim 3, wherein the Edit Order Line Items Process is **operable configured** to:

determine whether the status of the action code of an order line item is “Add”;
generate the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;
apply the current order line item to the complex asset;
display the currently requested state of the complex asset and allow the user to reconfigure the complex asset; and
generate the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset; and
ungroup any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1.

9. **(Currently Amended)** The complex order processing system of claim 3, wherein the Quote to Order Process is **operable configured** to:

ungroup any order line item that is tracked as an asset with a quantity of more than 1 into multiple line items, each with a quantity of 1
create a new order based on the quote;
set the order type of the new order to “Sales Order”;
set the order status of the new order to “New”; and
set the status of each order line item “New” if the action field is non-null.

10. **(Currently Amended)** The complex order processing system of claim 3, wherein the Update Opportunity Process is **operable configured** to:

- read the quote line items for a selected quote from the database.
- read the revenue items associated with any opportunity related to the selected quote;
- delete existing revenue items for the selected quote;
- create revenue items for each quote line item that has at least one of the following:
 - an action code of Add or Update;
 - a price type that is not Usage; and
 - an extended quantity that is not zero.

11. **(Currently Amended)** The complex order processing system of claim 3, wherein the Submit Order Process is **operable configured** to:

- determine whether the order status is not Open;
- an order header with account information;
- propagate the account information to line items in the order;
- set the status of the order to Open; and
- set the status of pending line items to Open.

12. **(Currently Amended)** The complex order processing system of claim 3, wherein the Apply Completed Sales Order Line Item To Service Profile Process is **operable configured** to:

- update an Order Item Processed flag of the topmost order line item based on whether all line items have a status of Complete.
- trim any incomplete or failed changes from the order line items whose products are not tracked as assets, line items whose status is not Complete, and line items whose action code is “-”;
- convert action codes of the order line items from Delete to Update;
- load the complex asset associated with the order line item from the database;
- create a new complex asset if there is no complex asset associated with the order line item;
- apply the order line item to the complex asset; and

store the complex asset.

13. **(Currently Amended)** The complex order processing system of claim 3, wherein the Move Process is **operable configured** to:

- identify a move-out address;
- identify a move-in address;
- retrieve the complex assets for the move-out address;
- generate a move-in quote with line items from the complex assets for the move-out address;
- add account information from the complex asset for the move-out address to the move-in quote;
- generate two quote line items for each quote line item in the move-out quote, wherein one quote line item has an action code of Delete and is associated with the move-out address, and the other quote line item has an action code of Add and is associated with the move-in address.
- specify a completed date for the move-out quote;
- assign a service point to each quote line item to be connected in the move-in quote; and
- specify a connect date for the move-in quote.

14. **(Currently Amended)** The complex order processing system of claim 2, wherein the Apply Method is **operable configured** to:

- retrieve a complex asset representing a customizable product;
- overlay unprocessed items and attributes of the customizable product from all of the open orders on the complex asset, wherein all of the open orders are represented in a property set in chronological order;
- generate a new property set representing the combination of the complex asset and the open orders; and
- set the hierarchy of the new property set to reflect the hierarchy of the open order.

15. **(Currently Amended)** The complex order processing system of claim 2, wherein the Delta Method is **operable configured** to:

determine when a change in the hierarchy occurs between a non-primary complex object and a primary complex object;
allow the user to indicate item fields and attributes in the two complex objects to be compared;
change the action code of an item to 'Add' when a new product is included in the non-primary complex object and not included in the primary complex object;
change the action code of an item to 'Delete' when a new product is included in the primary complex object and not included in the non-primary complex object;
change the action code of a child item to 'Update' when no other action is indicated;
change the action code of an item to 'Update' when the non-primary complex object's item and corresponding attribute(s) are added to the primary complex object; and
change the action code of the attribute of the item to:
 'Update' when the attribute data has changed;
 'Add' when the attribute has been added;
 'Delete' when the attribute has been deleted; and
 '-' (blank) when the attribute data has not changed,
when the action code of an item is changed to 'Update'.

16. **(Currently Amended)** The complex order processing system of claim 3, wherein the Trim Method is ~~operable~~ **configured** to:

 receive a complex object, wherein the complex object represents a quotes, orders, and a service profile;
 receive a trim specification, wherein the trim specification is a delta order
 begin with the top-most item in the complex object hierarchy, and proceeds recursively down through the root items to the trim any items and the children of the items that match criteria in the trim specification; and
 allow syntax for the trim specification to support arithmetic functions and logical operators.

17. **(Currently Amended)** The complex order processing system of claim 3, wherein the Explode Method is ~~operable~~ **configured** to:

change the quantity of the line item if the action code on the line item is 'Add'; and recognize a list of fields that should not be copied.

18. **(Currently Amended)** The complex order processing system of claim 3, wherein the Set Effectivity Dates Method is ~~operable~~ **configured** to:

generate the "Effective End Date" based on the date the action code of an item was set to "delete";

generate a "Calculated Effective Start Date" which is set to the date the action code of the item was set to "add"; and

generate a "Completed Date" based on the date the order was completed.

19. **(Currently Amended)** An order processing system comprising:

a complex object comprising at least one of[:]

a service profile,

quote information, and

order information pertaining to an account;

a data manager ~~operable~~ **configured** to

receive at least a portion of the service profile from an external server via an information network[;] and

generate a business object based on the portion of the service profile; and

a transformation engine ~~operable~~ **, wherein**

the transformation engine is configured to generate at least a portion of the complex object ~~based-on~~ **, and**

the transformation engine is further configured to perform the generation using the business object.

20. **(Currently Amended)** The order processing system of claim 19, further comprising:

user interface instructions ~~operable~~ **configured** to:

generate a customer portal view comprising a list of products and services

associated with the account, a list of quotes associated with the account, a

list of orders associated with the account, and a plurality of user-selectable options; and
receive data input from a user.

21. **(Currently Amended)** The order processing system of claim 19, further comprising:

a plurality of business service methods to perform ~~functions~~ **a function** associated with processing orders.

22. (Original) The order processing system of claim 21, further comprising:
a plurality of workflow processes, wherein each workflow process invokes at least one of the plurality of business service methods.

23. **(Currently Amended)** The order processing system of claim 20, further comprising:

instructions ~~operable~~ **configured** to update the status of order line items when a provisioning system completes all or part of an order.

a plurality of business service methods;

user interface instructions ~~operable~~ **configured** to:

generate a customer portal view comprising a list of products and services
associated with an account, a list of quotes associated with the account, a
list of orders associated with the account, and a plurality of user-selectable options;

receive input from a user; and

a plurality of workflow processes, wherein at least one of the plurality of workflow processes is invoked when a user selects one of the plurality of user-selectable options, and the at least one of the plurality of workflow processes invokes at least one of the plurality of business service methods to perform ~~functions~~ **a function** associated with the selected option.

24. (Original) The complex order processing system of claim 23, further comprising a plurality of complex objects, wherein each complex object is based on a property set of at least

one of: a complex asset, a quote, and an order; and the business service methods include at least one of:

- an Apply Method to create a hybrid asset/order that simulates the future configuration of the complex asset by overlaying unprocessed items and attributes from open orders on one of the plurality of complex objects that is based on the complex asset;
- a Delta Method to generate a delta complex object that represents the difference between at least two of the plurality of complex objects;
- a Trim Method to create a new complex object by eliminating all items that meet predefined trim criteria from one of the plurality of complex objects;
- a Logical Delete Method to generate a transformed delta order complex object by transforming items in one of the plurality of complex objects with an action code “Delete” to “Update” and changing status of the transformed items to “Inactive”;
- an Explode Method to copy items in one of the plurality of complex objects with quantity greater than one, and generate a delta quote complex object with a corresponding number of copies of the item with quantity equal to one;
- a Fully Exploded Method to determine whether the items in one of the plurality of complex objects have a quantity greater than one;
- a Field Delta Method to set the action code of a parent quote item to “Update” when a specified field is updated;
- a Set Effectivity Dates Method to generate a “Calculated Effective End Date”, a “Calculated Effective Start Date”, and a “Completed Date”, wherein the “Calculated Effective End Date” is set to the date the action code of an item in one of the plurality of complex objects based on an order was set to “delete”, the “Calculated Effective Start Date” is set to the date the action code of the item in the one of the plurality of complex objects was set to “add”, and the “Completed Date” is set to the date the order was completed;
- a Set Action Method to set the action code of an item in one of the plurality of complex objects;

- a Save Object Method to receive a key and a property set hierarchy that represents one of the plurality of complex objects, and saves the one of the plurality of complex objects in a string to object map using the key;
- a Get Object Method to receive the key and return the one of the plurality of complex objects that corresponds to the key;
- a Reset Method to delete all saved complex objects from the map;
- a Module Licensed Method to determine whether the complex order processing system is licensed for a particular computer system;
- an Update Order Status Method to check a Status field of a root item and all child items in the hierarchy of one of the plurality of complex objects based on a complex asset to determine whether the Status field is set to “Complete”, “Failed” or NULL;
- a Merge Method to copy all items from one of the plurality of complex objects to another of the plurality of complex objects;
- a Set Field Method to enumerate through the item hierarchy of one of the plurality of complex objects, find a named field of each item, and set the named field to a specified value; and
- a Create Empty Complex Object Method to create an empty complex object based on one of the plurality of complex objects.

25. **(Currently Amended)** The complex order processing system of claim 21, wherein the workflow processes comprise:

- a Supplemental Orders Process to create a new order by allowing the user to revise a pending order, wherein the Supplemental Orders Process is ~~operable~~ **configured** to:
 - generate the requested future state of the complex object by applying the prior active and unprocessed orders to the complex object;
 - apply a current order line item to the complex object;
 - allow the user to reconfigure the complex object;

generate the difference between the requested state of the complex object before the current order line item was applied and the state of the object after the user reconfigured the complex object; and
ungroup any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1.

26. (Original) The complex order processing system of claim 21, wherein the workflow processes comprise:

a Move Process to transfer a customer's complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection to transfer the complex asset configuration from the old location to the new location.

27. (Original) The complex order processing system of claim 21, further comprising a plurality of complex objects, wherein each complex object is based on a property set of at least one of: a complex asset, a quote, and an order; and the workflow processes comprise at least one of:

a New Products and Services Process to create a quote;
a Modify Products and Services Process to modify a complex asset in a quote;
a Disconnect Products and Services Process to disconnect a complex asset;
a Profile Process to retrieve account information that corresponds to a quote;
an Edit Quotes Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, and allow the user to make changes to the configuration of the complex asset and update a quote;
a Supplemental Orders Process to create a new order by allowing the user to revise a pending order;
an Edit Order Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, allow the user to make changes to the configuration of the complex asset, and update an order;
a Quote to Order Process to convert a quote to a sales order;
an Update Opportunity Process to create revenue items for all quote line items;

a Submit Order Process to associate account information with each line item in an order and set the status of the order to “open”;

an Ungroup Quotes Process to create multiple copies of line items in a quote that have quantities greater than one;

an Apply Completed Sales Order Line Item To Service Profile Process to update an asset table for a complex asset that has been provisioned; and

a Move Process to transfer a customer’s complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection.

28. **(Currently Amended)** The complex order processing system of claim 27, wherein the Modify Products and Services Process is **operable configured** to:

retrieve the quote that is selected;

create the quote if no quote is selected;

allow the user to make changes to the currently requested state of the complex asset;

apply changes specified by all open and pending orders related to the complex asset to simulate the state of the service product;

return an empty asset when the open and pending order disconnects a root component of the complex asset;

allow the user to change the configuration of the complex asset when the complex asset is customizable; and

update the quote with the change between the previously requested state of the complex asset and the newly requested state of the complex asset.

29. **(Currently Amended)** The complex order processing system of claim 27, wherein the Edit Quotes Line Items Process is **operable configured** to:

determine whether the status of the action code of an order line item is “Add”;

generate the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;

apply the current quote line item to the complex asset;

display the currently requested state of the complex asset and allow the user to reconfigure the complex asset; and
generate the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset.

30. **(Currently Amended)** The complex order processing system of claim 27, wherein the Edit Order Line Items Process is **operable configured** to:
determine whether the status of the action code of an order line item is “Add”;
generate the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;
apply the current order line item to the complex asset;
display the currently requested state of the complex asset and allow the user to reconfigure the complex asset; and
generate the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset; and
ungroup any line items in the complex asset with a quantity of more than 1 into multiple line items, each with a quantity of 1.

31. **(Currently Amended)** The complex order processing system of claim 27, wherein the Apply Completed Sales Order Line Item To Service Profile Process is **operable configured** to:
update an Order Item Processed flag of the topmost order line item based on whether all line items have a status of Complete.
trim any incomplete or failed changes from the order line items whose products are not tracked as assets, line items whose status is not Complete, and line items whose action code is “-”;
convert action codes of the order line items from Delete to Update;
load the complex asset associated with the order line item from the database;

create a new complex asset if there is no complex asset associated with the order line item; and
apply the order line item to the complex asset.

32. **(Currently Amended)** The complex order processing system of claim 27, wherein the Move Process is **operable configured** to:
identify a move-out address;
identify a move-in address;
retrieve the complex assets for the move-out address;
generate a move-in quote;
add account information from the complex asset for the move-out address to the move-in quote;
generate two quote line items for each quote line item in the move-out quote, wherein one quote line item has an action code of Delete and is associated with the move-out address, and the other quote line item has an action code of Add and is associated with the move-in address;
specify a completed date for the move-out quote; and
assign a service point to each quote line item to be connected in the move-in quote.

33. **(Currently Amended)** The complex order processing system of claim 24, wherein the Apply Method is **operable configured** to:
instructions to retrieve a complex asset representing a customizable product;
instructions to overlay unprocessed items and attributes of the customizable product from all of the open orders on the complex asset, wherein all of the open orders are represented in a property set in chronological order;
instructions to generate a new property set representing the combination of the complex asset and the open orders; and
instructions to set the hierarchy of the new property set to reflect the hierarchy of the open order.

34. **(Currently Amended)** The complex order processing system of claim 24, wherein the Delta Method is **operable configured** to:

determine when a change in the hierarchy occurs between a non-primary complex object and a primary complex object;
allow the user to indicate item fields and attributes in the two complex objects to be compared;
change the action code of an item to 'Add' when a new product is included in the non-primary complex object and not included in the primary complex object;
change the action code of an item to 'Delete' when a new product is included in the primary complex object and not included in the non-primary complex object;
change the action code of a child item to 'Update' when no other action is indicated;
change the action code of an item to 'Update' when the non-primary complex object's item and corresponding attribute(s) are added to the primary complex object; and
change the action code of the attribute of the item to:
 'Update' when the attribute data has changed;
 'Add' when the attribute has been added;
 'Delete' when the attribute has been deleted; and
 '-' (blank) when the attribute data has not changed,
when the action code of an item is changed to 'Update'.

35. **(Currently Amended)** The complex order processing system of claim 24, wherein the Trim Method is ~~operable~~ **configured** to:
 receive a complex object, wherein the complex object represents a quotes, orders, and a service profile;
 receive a trim specification, wherein the trim specification is a delta order
 begin with the top-most item in the complex object hierarchy, and proceeds recursively down through the root items to the trim any items and the children of the items that match criteria in the trim specification; and
 allow syntax for the trim specification to support arithmetic functions and logical operators.

36. **(Currently Amended)** An apparatus for processing orders comprising:
 a plurality of business service methods;

means for generating a customer portal ~~comprising~~ , wherein

the means for generating a customer portal comprises

a list of products and services associated with an account,

a list of quotes associated with the account,

a list of orders associated with the account, and

a plurality of user-selectable options, ~~wherein~~

the customer portal is accessible by ~~users including a user~~ , wherein

the user is one of a customer service representative[[s]], a customer[[s]],

and a service provider[[s]] for the customer[[s]];

means for receiving input from [[a]] the user; ~~and~~

means for invoking a workflow process, [[and]] wherein

the means for invoking the workflow process ~~invokes~~ comprises a means for

invoking at least one of the plurality of business service methods, and

the at least one of the plurality of business service methods is configured to

perform ~~functions~~ a function associated with the selected option;

means for receiving at least one of[[:]]

a service profile,

quote information, and

order information pertaining to an account from an external server;

means for generating a business object ~~based on~~ , wherein

means for generating uses the at least one of[[:]]

[[a]] the service profile,

the quote information, and

the order information; and

means for transforming at least a portion of the business object [[to]] into a complex object.

37. (Currently Amended) The apparatus of claim 36, further comprising:

instructions ~~operable~~ configured to update the status of order line items when a provisioning system completes all or part of an order.

38. (Original) The apparatus of claim 36, wherein the plurality of business service methods include at least one of:

- an Apply Method to create a hybrid asset/order that simulates the future configuration of the complex asset by overlaying unprocessed items and attributes from open orders on one of the plurality of complex objects that is based on the complex asset;
- a Delta Method to generate a delta complex object that represents the difference between at least two of the plurality of complex objects;
- a Trim Method to create a new complex object by eliminating all items that meet predefined trim criteria from one of the plurality of complex objects;
- a Logical Delete Method to generate a transformed delta order complex object by transforming items in one of the plurality of complex objects with an action code “Delete” to “Update” and changing status of the transformed items to “Inactive”;
- an Explode Method to copy items in one of the plurality of complex objects with quantity greater than one, and generate a delta quote complex object with a corresponding number of copies of the item with quantity equal to one;
- a Fully Exploded Method to determine whether the items in one of the plurality of complex objects have a quantity greater than one;
- a Field Delta Method to set the action code of a parent quote item to “Update” when a specified field is updated;
- a Set Effectivity Dates Method to generate a “Calculated Effective End Date”, a “Calculated Effective Start Date”, and a “Completed Date”, wherein the “Calculated Effective End Date” is set to the date the action code of an item in one of the plurality of complex objects based on an order was set to “delete”, the “Calculated Effective Start Date” is set to the date the action code of the item in the one of the plurality of complex objects was set to “add”, and the “Completed Date” is set to the date the order was completed;
- a Set Action Method to set the action code of an item in one of the plurality of complex objects;

- a Save Object Method to receive a key and a property set hierarchy that represents one of the plurality of complex objects, and saves the one of the plurality of complex objects in a string to object map using the key;
- a Get Object Method to receive the key and return the one of the plurality of complex objects that corresponds to the key;
- a Reset Method to delete all saved complex objects from the map;
- a Module Licensed Method to determine whether the complex order processing system is licensed for a particular computer system;
- an Update Order Status Method to check a Status field of a root item and all child items in the hierarchy of one of the plurality of complex objects based on a complex asset to determine whether the Status field is set to “Complete”, “Failed” or NULL;
- a Merge Method to copy all items from one of the plurality of complex objects to another of the plurality of complex objects;
- a Set Field Method to enumerate through the item hierarchy of one of the plurality of complex objects, find a named field of each item, and set the named field to a specified value; and
- a Create Empty Complex Object Method to create an empty complex object based on one of the plurality of complex objects.

39. (Original) The apparatus of claim 36, wherein the workflow process comprises:
- a Supplemental Orders Process to create a new order by allowing the user to revise a pending order, wherein the Supplemental Orders Process comprises:
 - means to generate the requested future state of the complex object by applying the prior active and unprocessed orders to the complex object;
 - means to apply a current order line item to the complex object;
 - means to allow the user to reconfigure the complex object; and
 - means to generate the difference between the requested state of the complex object before the current order line item was applied and the state of the object after the user reconfigured the complex object; and

means to ungroup any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1..

40. (Original) The apparatus of claim 36, wherein the workflow process comprises: a Move Process to transfer a customer's complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection to transfer the complex asset configuration from the old location to the new location.

41. (Original) The apparatus of claim 36, wherein the workflow process comprises at least one of:

- a New Products and Services Process to create a quote;
- a Modify Products and Services Process to modify a complex asset in a quote;
- a Disconnect Products and Services Process to disconnect a complex asset;
- a Profile Process to retrieve account information that corresponds to a quote;
- an Edit Quotes Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, and allow the user to make changes to the configuration of the complex asset and update a quote;
- a Supplemental Orders Process to create a new order by allowing the user to revise a pending order;
- an Edit Order Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, allow the user to make changes to the configuration of the complex asset, and update an order;
- a Quote to Order Process to convert a quote to a sales order;
- an Update Opportunity Process to create revenue items for all quote line items;
- a Submit Order Process to associate account information with each line item in an order and set the status of the order to "open";
- an Ungroup Quotes Process to create multiple copies of line items in a quote that have quantities greater than one;
- an Apply Completed Sales Order Line Item To Service Profile Process to update an asset table for a complex asset that has been provisioned; and

a Move Process to transfer a customer's complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection.

42. (Original) The apparatus of claim 41, wherein the Modify Products and Services Process comprises:

- means for retrieving the quote that is selected;
- means for creating the quote if no quote is selected;
- means for allowing the user to make changes to the currently requested state of the complex asset;
- means for applying changes specified by all open and pending orders related to the complex asset to simulate the state of the service product;
- means for returning an empty asset when the open and pending order disconnects a root component of the complex asset;
- means for allowing the user to change the configuration of the complex asset when the complex asset is customizable;
- means for updating the quote with the change between the previously requested state of the complex asset and the newly requested state of the complex asset; and
- means for storing the updated quote.

43. (Original) The apparatus of claim 41, wherein the Edit Quotes Line Items Process comprises:

- means for determining whether the status of the action code of an order line item is "Add";
- means for generating the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;
- means for applying the current quote line item to the complex asset;
- means for allowing the user to reconfigure the complex asset; and

means for generating the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset.

44. (Original) The apparatus of claim 41, wherein the Edit Order Line Items Process comprises:

means for determining whether the status of the action code of an order line item is “Add”;

means for generating the requested future state of the complex asset by applying the prior active and unprocessed orders to the complex asset;

means for applying the current order line item to the complex asset;

means for displaying the currently requested state of the complex asset and allow the user to reconfigure the complex asset; and

means for generating the difference between the requested state of the complex asset before the current order line item was applied and the state of the asset after the user reconfigured the complex asset; and

means for ungrouping any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1.

45. (Original) The apparatus of claim 41, wherein the Apply Completed Sales Order Line Item To Service Profile Process comprises:

means for updating an Order Item Processed flag of the topmost order line item based on whether all line items have a status of Complete.

means for trimming any incomplete or failed changes from the order line items whose products are not tracked as assets, line items whose status is not Complete, and line items whose action code is “-”;

means for converting action codes of the order line items from Delete to Update;

means for loading the complex asset associated with the order line item from the database;

means for creating a new complex asset if there is no complex asset associated with the order line item;

means for applying the order line item to the complex asset; and
means for storing the complex asset.

46. (Original) The apparatus of claim 41, wherein the Move Process comprises:
means for identifying a move-out address;
means for identifying a move-in address;
means for retrieving the complex assets for the move-out address;
means for generating a move-in quote with line items from the complex assets for the
move-out address;
means for adding account information from the complex asset for the move-out address
to the move-in quote;
means for generating two quote line items for each quote line item in the move-out quote,
wherein one quote line item has an action code of Delete and is associated with
the move-out address, and the other quote line item has an action code of Add and
is associated with the move-in address.
means for specifying a completed date for the move-out quote;
means for assigning a service point to each quote line item to be connected in the move-
in quote; and
means for specifying a connect date for the move-in quote.

47. (Original) The apparatus of claim 38, wherein the Apply Method comprises:
means for retrieving a complex asset representing a customizable product;
means for overlaying unprocessed items and attributes of the customizable product from
all of the open orders on the complex asset, wherein all of the open orders are
represented in a property set in chronological order;
means for generating a new property set representing the combination of the complex
asset and the open orders; and
means for setting the hierarchy of the new property set to reflect the hierarchy of the open
order.

48. (Original) The apparatus of claim 38, wherein the Delta Method comprises:

means for determining when a change in the hierarchy occurs between a non-primary complex object and a primary complex object;

means for allowing the user to indicate item fields and attributes in the two complex objects to be compared;

means for changing the action code of an item to 'Add' when a new product is included in the non-primary complex object and not included in the primary complex object;

means for changing the action code of an item to 'Delete' when a new product is included in the primary complex object and not included in the non-primary complex object;

means for changing the action code of a child item to 'Update' when no other action is indicated;

means for changing the action code of an item to 'Update' when the non-primary complex object's item and corresponding attribute(s) are added to the primary complex object; and

means for changing the action code of the attribute of the item to:

- 'Update' when the attribute data has changed;
- 'Add' when the attribute has been added;
- 'Delete' when the attribute has been deleted; and
- '-' (blank) when the attribute data has not changed,

when the action code of an item is changed to 'Update'.

49. (Original) The apparatus of claim 38, wherein the Trim Method comprises:

means for receiving a complex object, wherein the complex object represents a quotes, orders, and a service profile;

means for receiving a trim specification, wherein the trim specification is a delta order

means for beginning with the top-most item in the complex object hierarchy, and proceeds recursively down through the root items to the trim any items and the children of the items that match criteria in the trim specification; and

means for allowing syntax for the trim specification to support arithmetic functions and logical operators.

50. (Currently Amended) A method for processing orders comprising:
generating a customer portal ~~comprising~~ , wherein

the generating a customer portal comprises

a list of products and services associated with an account,

a list of quotes associated with the account,

a list of orders associated with the account, and

a plurality of user-selectable options, ~~wherein and~~

the customer portal is accessible by ~~users including a user~~, wherein

the user is one of a customer service representative[[s]], a customer[[s]],

and a service provider[[s]] for the customer[[s]];

invoking a workflow process, ~~[[and]]~~ wherein

the workflow process invokes at least one business service method ~~to perform~~

functions, and

the at least one business service method performs a function associated with

the selected option;

receiving at least one of:

a service profile,

quote information, and

order information pertaining to an account;

generating a business object, wherein

the generating is based on the at least one of[[:]]

[[a]] the service profile,

the quote information, and

the order information; and

transforming at least a portion of the business object ~~[[to]]~~ into a complex object.

51. (Currently Amended) The method of claim 50, further comprising:

instructions ~~operable~~ configured to update the status of order line items when a provisioning system completes all or part of an order.

52. (Original) The method of claim 50, wherein the plurality of business service methods include at least one of:

- an Apply Method to create a hybrid asset/order that simulates the future configuration of the complex object by overlaying unprocessed items and attributes from open orders on one of the plurality of complex objects that is based on the complex object;
- a Delta Method to generate a delta complex object that represents the difference between at least two of the plurality of complex objects;
- a Trim Method to create a new complex object by eliminating all items that meet predefined trim criteria from one of the plurality of complex objects;
- a Logical Delete Method to generate a transformed delta order complex object by transforming items in one of the plurality of complex objects with an action code “Delete” to “Update” and changing status of the transformed items to “Inactive”;
- an Explode Method to copy items in one of the plurality of complex objects with quantity greater than one, and generate a delta quote complex object with a corresponding number of copies of the item with quantity equal to one;
- a Fully Exploded Method to determine whether the items in one of the plurality of complex objects have a quantity greater than one;
- a Field Delta Method to set the action code of a parent quote item to “Update” when a specified field is updated;
- a Set Effectivity Dates Method to generate a “Calculated Effective End Date”, a “Calculated Effective Start Date”, and a “Completed Date”, wherein the “Calculated Effective End Date” is set to the date the action code of an item in one of the plurality of complex objects based on an order was set to “delete”, the “Calculated Effective Start Date” is set to the date the action code of the item in the one of the plurality of complex objects was set to “add”, and the “Completed Date” is set to the date the order was completed;
- a Set Action Method to set the action code of an item in one of the plurality of complex objects;

- a Save Object Method to receive a key and a property set hierarchy that represents one of the plurality of complex objects, and saves the one of the plurality of complex objects in a string to object map using the key;
- a Get Object Method to receive the key and return the one of the plurality of complex objects that corresponds to the key;
- a Reset Method to delete all saved complex objects from the map;
- a Module Licensed Method to determine whether the complex order processing system is licensed for a particular computer system;
- an Update Order Status Method to check a Status field of a root item and all child items in the hierarchy of one of the plurality of complex objects based on a complex object to determine whether the Status field is set to "Complete", "Failed" or NULL;
- a Merge Method to copy all items from one of the plurality of complex objects to another of the plurality of complex objects;
- a Set Field Method to enumerate through the item hierarchy of one of the plurality of complex objects, find a named field of each item, and set the named field to a specified value; and
- a Create Empty Complex Object Method to create an empty complex object based on one of the plurality of complex objects.

53. (Original) The method of claim 50, wherein the workflow process comprises:

a Supplemental Orders Process to create a new order by allowing the user to revise a pending order, wherein the Supplemental Orders Process comprises:

means to generate the requested future state of the complex object by applying the prior active and unprocessed orders to the complex object;

means to apply a current order line item to the complex object;

means to allow the user to reconfigure the complex object; and

means to generate the difference between the requested state of the complex object before the current order line item was applied and the state of the object after the user reconfigured the complex object; and

means to ungroup any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1..

54. **(Currently Amended)** The ~~method~~ apparatus of claim 36, wherein the workflow process comprises:

a Move Process to transfer a customer's asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection to transfer the asset configuration from the old location to the new location.

55. **(Original)** The method of claim 50, wherein the workflow process comprises at least one of:

a New Products and Services Process to create a quote;
a Modify Products and Services Process to modify a complex asset in a quote;
a Disconnect Products and Services Process to disconnect a complex asset;
a Profile Process to retrieve account information that corresponds to a quote;
an Edit Quotes Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, and allow the user to make changes to the configuration of the complex asset and update a quote;
a Supplemental Orders Process to create a new order by allowing the user to revise a pending order;
an Edit Order Line Items Process to apply all active and unprocessed orders to a complex asset to generate the future state of the complex asset, allow the user to make changes to the configuration of the complex asset, and update an order;
a Quote to Order Process to convert a quote to a sales order;
an Update Opportunity Process to create revenue items for all quote line items;
a Submit Order Process to associate account information with each line item in an order and set the status of the order to "open";
an Ungroup Quotes Process to create multiple copies of line items in a quote that have quantities greater than one;

an Apply Completed Sales Order Line Item To Service Profile Process to update an asset table for a complex asset that has been provisioned; and
a Move Process to transfer a customer's complex asset configuration from an old location to a new location by entering the address of the new location and entering a single option selection.

56. (Original) The method of claim 55, wherein the Modify Products and Services Process comprises:

- retrieving the quote that is selected;
- creating the quote if no quote is selected;
- allowing the user to make changes to the currently requested state of the complex object;
- applying changes specified by all open and pending orders related to the complex object to simulate the state of the service product;
- returning an empty asset when the open and pending order disconnects a root component of the complex object;
- allowing the user to change the configuration of the complex object when the complex object is customizable; and
- updating the quote with the change between the previously requested state of the complex object and the newly requested state of the complex object.

57. (Original) The method of claim 55, wherein the Edit Quotes Line Items Process comprises:

- determining whether the status of the action code of an order line item is "Add";
- generating the requested future state of the complex object by applying the prior active and unprocessed orders to the complex object;
- applying the current quote line item to the complex object;
- allowing the user to reconfigure the complex object; and
- generating the difference between the requested state of the complex object before the current order line item was applied and the state of the asset after the user reconfigured the complex object.

58. **(Currently Amended)** The method of claim 55, wherein the Edit Order Line Items Process comprises:

determining whether the status of the action code of an order line item is “Add”;
generating the requested future state of the complex object by applying the prior active and unprocessed orders to the complex object;
applying the current order line item to the complex object;
displaying the currently requested state of the complex object and allow the user to reconfigure the complex object; **[[and]]**
generating the difference between the requested state of the complex object before the current order line item was applied and the state of the asset after the user reconfigured the complex object; and
ungrouping any order line item with a quantity of more than 1 into multiple line items, each with a quantity of 1.

59. **(Original)** The method of claim 55, wherein the Apply Completed Sales Order Line Item To Service Profile Process comprises:

updating an Order Item Processed flag of the topmost order line item based on whether all line items have a status of Complete.
trimming any incomplete or failed changes from the order line items whose products are not tracked as assets, line items whose status is not Complete, and line items whose action code is “-”;
converting action codes of the order line items from Delete to Update;
loading the complex object associated with the order line item from the database;
creating a new complex object if there is no complex object associated with the order line item;
applying the order line item to the complex object; and
storing the complex object.

60. **(Original)** The method of claim 55, wherein the Move Process comprises:
identifying a move-out address;
identifying a move-in address;

retrieving the complex objects for the move-out address;
generating a move-in quote with line items from the complex objects for the move-out address;
adding account information from the complex object for the move-out address to the move-in quote;
generating two quote line items for each quote line item in the move-out quote, wherein one quote line item has an action code of Delete and is associated with the move-out address, and the other quote line item has an action code of Add and is associated with the move-in address.
specifying a completed date for the move-out quote;
assigning a service point to each quote line item to be connected in the move-in quote;
and
specifying a connect date for the move-in quote.

61. (Original) The method of claim 52, wherein the Apply Method comprises:
retrieving a complex object representing a customizable product;
overlaying unprocessed items and attributes of the customizable product from all of the open orders on the complex object, wherein all of the open orders are represented in a property set in chronological order;
generating a new property set representing the combination of the complex object and the open orders; and
setting the hierarchy of the new property set to reflect the hierarchy of the open order.

62. (Original) The method of claim 52, wherein the Delta Method comprises:
determining when a change in the hierarchy occurs between a non-primary complex object and a primary complex object;
allowing the user to indicate item fields and attributes in the two complex objects to be compared;
changing the action code of an item to 'Add' when a new product is included in the non-primary complex object and not included in the primary complex object;

changing the action code of an item to 'Delete' when a new product is included in the primary complex object and not included in the non-primary complex object; changing the action code of a child item to 'Update' when no other action is indicated; changing the action code of an item to 'Update' when the non-primary complex object's item and corresponding attribute(s) are added to the primary complex object; and changing the action code of the attribute of the item to:

- 'Update' when the attribute data has changed;
- 'Add' when the attribute has been added;
- 'Delete' when the attribute has been deleted; and
- '-' (blank) when the attribute data has not changed,

when the action code of an item is changed to 'Update'.

63. (Original) The method of claim 52, wherein the Trim Method comprises:

receiving a complex object, wherein the complex object represents a quotes, orders, and a service profile;

receiving a trim specification, wherein the trim specification is a delta order beginning with the top-most item in the complex object hierarchy, and proceeds recursively down through the root items to the trim any items and the children of the items that match criteria in the trim specification; and

allowing syntax for the trim specification to support arithmetic functions and logical operators.

64-66. (Cancelled)